Introduction (Analysis of Speech Signal Spectrums Using the L2 Norm)
Introduction to Speech Analysis Project

I – Introduction

Signals analysis is commonly applied to voice signals. Voice signals are modulated using Amplitude Modulation (AM) for broadcast distributions, and signals are compressed before transmission when speaking into a phone. This report will introduce the use of the Fourier Transform into speech signals analysis. Specifically, we will attempt to use the Fourier Transform to identify the speaker of a series of words.

To focus our efforts, we present the following problem:

Biometric identification has begun to be used to maintain security. Fingerprint identification door locks are now sold through common stores like Staples[footnote]. A development team in Spain has created an IRIS scanner that performs person verification to unlock cellular phones using the phone's built in camera[footnote]. Many biometric identification sensors are expensive, and therefore they have yet to proliferate. A common microphone, though, is very cheap – even cheaper than a camera. This paper investigates the possibility of using voice signals to perform identification.
http://www.staples.com/iTouchless-Bio- Matic-Fingerprint-Door-Lock-Gold-Left-Handle/product_799740? cmArea=SEARCH
http://www.biowallet.net/ind ex.php? option=com_content&view=frontpage&Itemid=1

Specifically, we expect a user to say four numbers from the set "zero" through "nine". These four numbers are the user's Personal Identification Number (PIN). We will make a comparison of each stated number with previous recordings of that user, and use those comparisons to make a decision as to whether or not there is a match to the user.

II – Audacity

This section of the report fulfills the Audacity assignment located here: http://moodle.csun.edu/mod/assignment/view.php?id=2772.

For this entire project, sounds were recorded using the Audacity software and a Veo Advanced Connect Web Cam / Microphone. The phrase recorded was Nicholas stating “Zero, one, two”.

Table 2.1 shows the size of the resulting files when recorded at the specified sampling rates.

Sampling Rate (Fs)	Size (bytes)	Size / (Size of 8000)	Fs / (Fs of 8000)
8,000	26,794	1	1
16,000	49,456	1.8	2
44,100	152,620	5.7	5.5

Table 2.1: Statistics for file recorded at various sampling rates.

Notice that the ratio of the file sizes does not correspond exactly to the ratio of sampling frequencies. This is because, although the sequences represent the same phrase, each sequence took a different amount of time to state.

User Manual (Analysis of Speech Signal Spectrums Using the L2 Norm)
User Manual for Speech Analysis Project

III – User Manual

The software for this project was written in Matlab. The main codes is voiceRecognition.m. The prototype for the voiceRecognition functions is as follows:

trueMatch = voiceRecognition(username, pin, thresh,

candidateName)

It is assumed that each user has a username. In the files packaged with this report, the two users present in the database are ‘Nicholas’ and ‘Andrew’[footnote]. The username parameter for the voiceRecognition is a 1D character array with characters equal to the username.
Andrew is Nicholas’ roommate and was kind enough to allow himself to be recorded for the purposes of this project.

The pin parameter represents the user’s PIN. It is a four element 1D array containing integer values 0 – 9.

thresh is an optional parameter. There is a threshold associated with the final matching algorithm (discussed later). By default, this threshold is set to 0.48; but it can be changed by passing a double value between 0 and 1 into this parameter.

candidateName is an optional parameter. The candidate is the person whose recording will be compared to the username’s database. In practice, the candidate name and the username would always be the same. However, for testing purposes, it was convenient to be able to specify a different candidate – e.g. we would specify the username as ‘Nicholas’ and the candidate as ‘Andrew’ to verify that the software did indeed detect an impostor.

(3.1) shows an example of a voice recognition function call.

voiceRecognition('Nicholas' , [0,1,2,3]);	(3.1)

Directory Structure

The voiceRecognition software assumes a specific directory structure. There is a directory called ‘recordings’ in the same location as the Matlab working directory. Within the recordings directory are two directories: ‘current’ and ‘person’. The person directory contains the database of users who have previously entered their data into the system – i.e. it is the database of valid users. The current directory contains recordings of candidates.

Within the person directory, each user is granted his/her own directory. For example, there is a directory present named ‘Nicholas’. Within ‘Nicholas’ there should be seven wav file recordings of each pin number. For example, if the pin were 1-8-1-9, then there should be seven recordings of “one”, “eight”, and “nine”. The recordings must be labeled *num*1.wav through *num*7.wav.

Within the current directory, there again should be a directory for each candidate that would be presented to the software. In our case, we present no other candidates than those included in the user database. Within this directory, there must be recordings for all PIN numbers. In the above example, there should be present the files one.wav, eight.wav, and nine.wav.

Assumptions

There can be a significant amount of variability in the frequencies that a person uses to generate a word. The assumption made in this software is that the person generates the word attempting to use the same pitch and speed as previously conducted when submitting a phrase for comparison.

We now go on to describe the algorithms implemented in the software.

Overall Approach (Analysis of Speech Signal Spectrums Using the L2 Norm)
This module describes the overall approach.

IV – Overall Approach

The overall approach is shown in figure 4.1. A PIN consists of a username and recordings of four separate (though not necessarily unqiue) numbers. A user is added to the database by adding seven recordings of all PIN numbers.

A candidate is a specified username and a single recoding of all PIN numbers. Upon submission to the software, the relevant signal is extracted from the entire candidate recording.

From the database, all seven recordings are combined into an “average” signal. The average signal and the extracted candidate signal are the compared – this results in a metric value. Based on this metric, a decision is made as to whether or not the candidate is a match or an impostor.

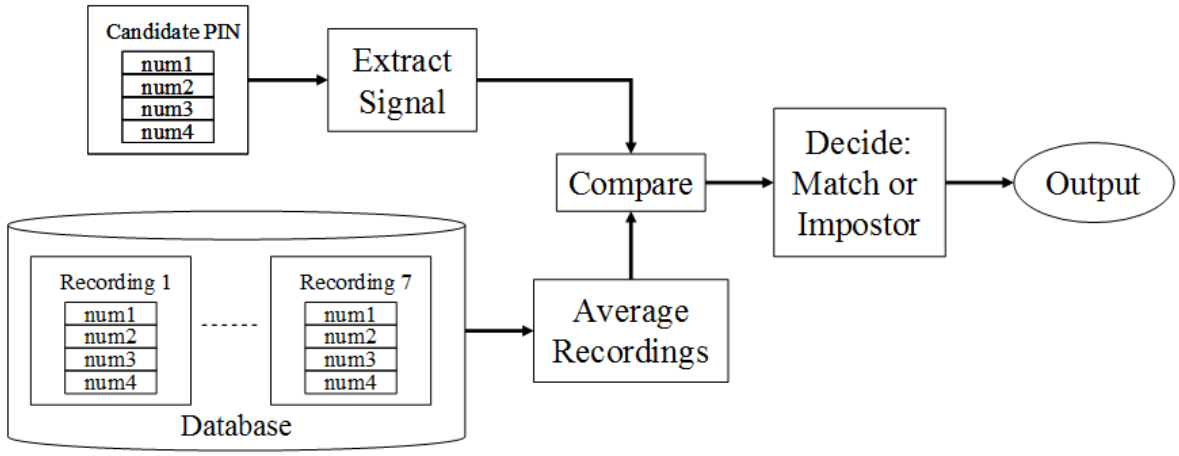

Figure 4.1: Overall approach for speaker identification

Signal Extraction (Analysis of Speech Signal Spectrums Using the L2 Norm)
This section describes how the signal is extracted.

V – Signal Extraction

Prior to signal comparison, the signals must first be aligned. The first step to alignment is to extract the relevant signal from the entire data segment. To perform this initial processing we smooth the absolute value of the data, and find the maximum of the smoothed data. Given the index of the maximum, we extend the bounds of our hypothesized signal outwards until the amount of energy within our bounds exceeds a threshold percentage. Energy is defined as the L2 norm of the data signal, shown in (5.1). This procedure is encapsulated in the function “extractSignal”.

$E = \sum_i \mathrm{data}(i)^2$	(5.1)

The threshold percentage was determined empirically. Figure 5.1 shows examples of the signal that was extracted for different threshold percentages. Portions of the data that were edited out are replaced with 0. As one can see, thresholding at 90% removes relevant information from the data, and thresholding at 99% retains much of the irrelevant values in the data. We set the energy threshold to 95%.

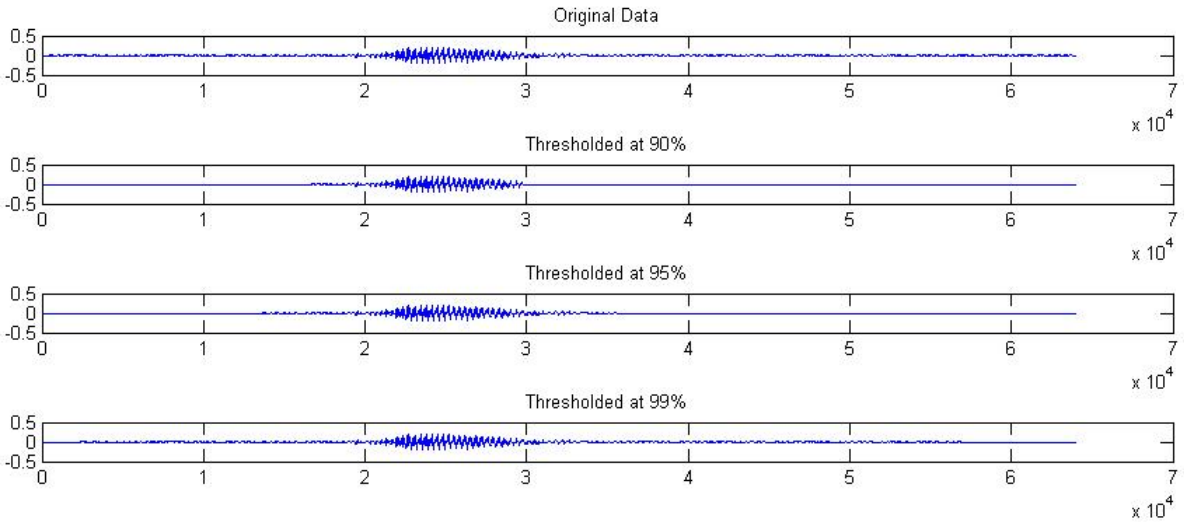

Figure 5.1: Results of identifying the signal in the data segment based on different energy percentage thresholds. The original data shown is of Nicholas stating the word “one”. Portions that are removed from the signal are set to 0.

Class Tutorial (Analysis of Speech Signal Spectrums Using the L2 Norm) This method describes the course tutorial supplied by the professor.

VI – Class Tutorial

This section is based on the class tutorial stated here: http://moodle.csun.edu/file.php/177/VoiceRecognition/node5.html.

For this tutorial, we analyze the sequence shown in figure 6.1; this figure shows two recordings of Nicholas saying the word “two”. We denote these sequences as two1 and two2.

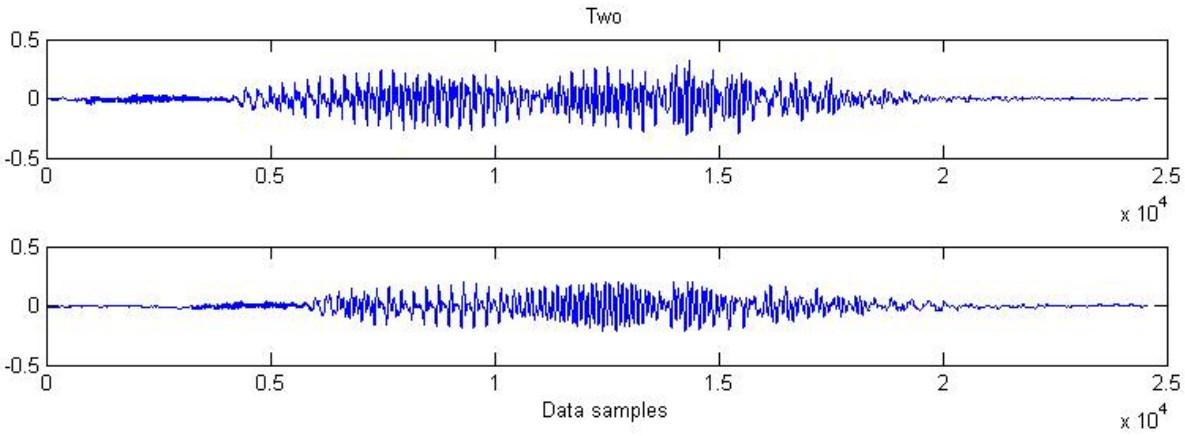

Figure 6.1: Two recordings of Nicholas saying the word “two”.

We first compute the L2 norm of the difference of two signals as shown in (6.1).

$$\| f_1 - f_2 \| = \sum_i^{\min(N_1,N_2)} (f_1(i) - f_2(i))^2 \quad (6.1)$$

We naively cut off the comparison of the two data sequences when the shorter signal ends. The norm of the difference between these two sequences is approximately 15.4. To gain an understanding of whether this value is large, we compute the energy in the individual signals. The energy in two1 and two2 are approximately 12.0 and 9.3, respectively. We see that the norm of the difference is greater than 100% of the energy in each individual signal. This is very large for two signals that produce the same sound (where “same” here means that both signals are interpreted by a human as having the same meaning).

We now compare the norm of the first “two” sequence to itself. Shown in figure 6.2 are two sequences: two1 and two3, where two3 = 5 * two1. Note the difference in the values of the y axes. As one can see, the difference in the signals is large (as was expected).

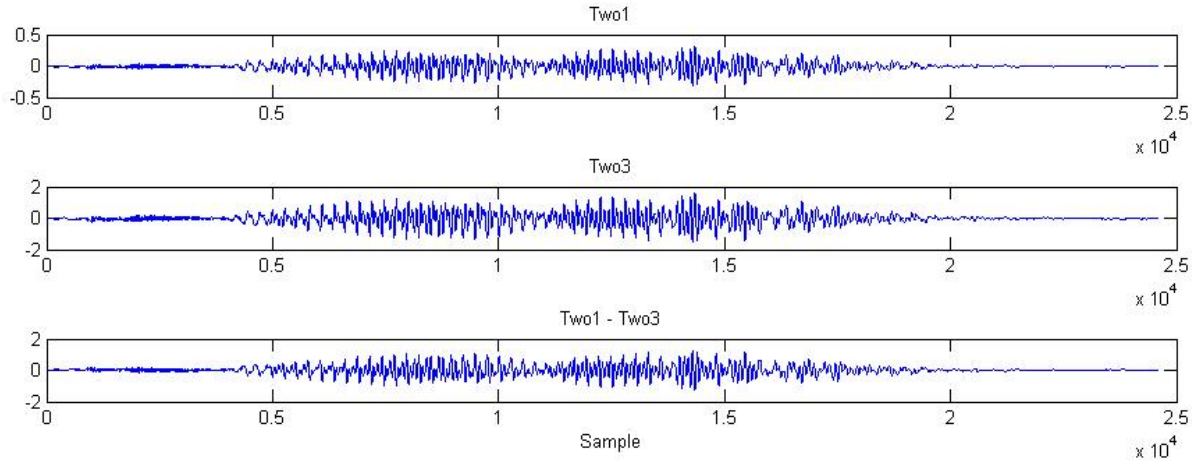

Figure 6.2: Plots showing a sequence “two” stated by Nicholas, that signal multiplied by 5, and the difference of the two.

Were two1 and two3 different recordings of the same person saying the phrase “two”, we could first make the sequences comparable by normalizing the amount the two sequences. As suggested in the tutorial, we could normalize by the maximum value in the signal. This is done according to the formula shown in (6.2).

$$\text{normalized data} = \frac{\text{data}}{\max(\text{data})}$$	(6.2)

In this case this procedure works perfectly, and in fact the L2 norm of the difference vector between two1 and the normalized two3 is 0. However, this procedure only works because one signal is exactly a multiple of the other. If the signals were slightly misaligned, or if there were noise added to the signal, then the energy in the difference signal would again be on the order of the energy in the signal itself. There would not have to be a lot of noise to corrupt this procedure. If two3 equaled 5*two1 at all points except the maximum, and that point were corrupted such that it were 2*5*two1, then the average value for the ratio between the two1 and the normalized data would be approximately 2.

A more robust normalization procedure is to normalize by the energy in the signal. This is done according to the formula shown in (6.3); the 2 subscript denotes that the 2 norm is used.

$$\text{normalized data} = \frac{\text{data}}{\|\text{data}\|_2}$$	(6.3)

Though this procedure does not make the comparison robust to alignment issues, it does make the procedure slightly robust to spurious noise, as long as that noise has a 0 temporal mean. Again, in our example where no noise is added to the system and the signals are perfectly aligned, the L2 norm of the difference between two1 and the normalized two3 is 0.

Comparing the norms as performed above is interesting; this procedure reveals just how adaptable the human brain is. The same phrase emitted by the same person while changing the amount of contraction in the diaphragm, the amount of contraction of the intercostals muscles, the spectrum emitted by the vocal cords (changing the pitch), and the shape of

the respiratory tract (e.g. the shape of the mouth) are easily interpreted by the human brain to have the same meaning.

For a computer to perform similarly, we will need a more sophisticated processing than a comparison of norms.

Spectral Comparison (Analysis of Speech Signal Spectrums Using the L2 Norm)
Initial publication of module.

VII – Spectral Comparison Using the L2 Norm

A common way to determine similarity is to compute the normalized correlation between two signals (as shown in (7.1)); here, d represents the data segment, σ^2 represents the variance of the signal, and γ is the normalized correlation value[footnote]. The multiplication of the demeaned data segments is an element by element multiplication. A common correlation threshold used for similarity in signal processing applications is 95%. It is interesting to note that the normalized correlation value for two1 and two2 is approximately 32%; this value is remarkably low.
Lewis, J.P., Fast Normalized Cross Correlation, *Vision Interface*, 1995, pp. 120-123

$\gamma = \frac{\left(d_1 - \bar{d}_1\right)\left(d_2 - \bar{d}_2\right)}{\sqrt{\sigma_1^2 \cdot \sigma_2^2}}$	(7.1)

We get of hint of the similarities by observing a spectrogram of the two signals, shown in figure 7.1. By eye, we see that the two signals show similar spectral content through the phrase. The “trick” will be getting the computer to recognize that these two sequences are the same, as our eye does.

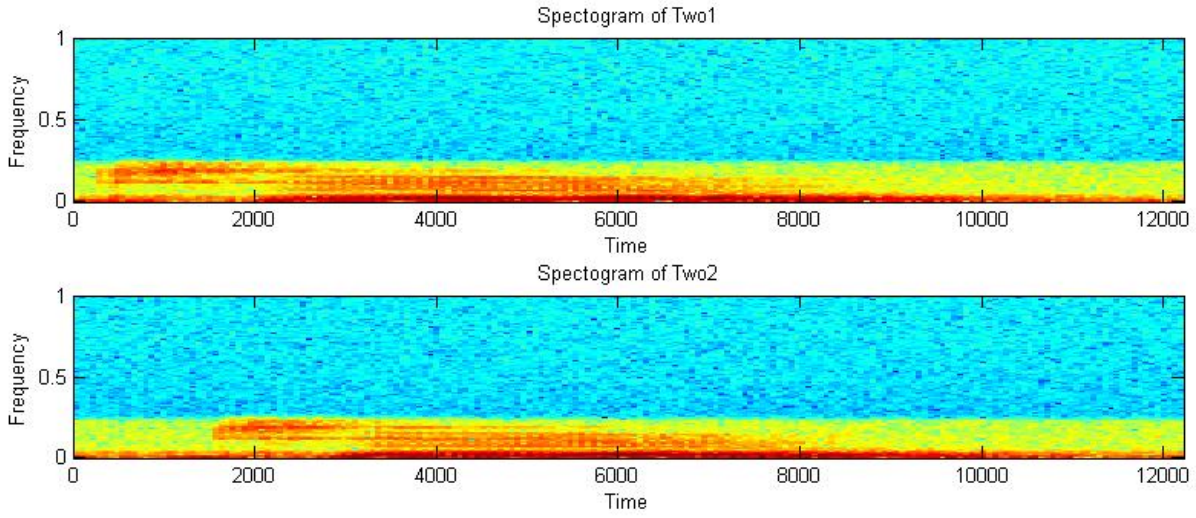

Figure 7.1: Spectograms of Two1 and Two2.

We begin by computing the norm of the difference of the spectrums. The procedure for doing so is shown in figure 7.2. We analyze this procedure step by step below.

[missing_resource: graphics2.wmf]

Figure 7.2: Procedure for computing the norm of the difference of the two signal spectrums.

N_{max} is the maximum of the number of samples in the two data segments; i.e. $N_{\max} = \max(\text{length1},\text{length2})$. We zero pad the shorter signal such that it is the same length as the longer segment. We then calculate the FFT of the two zero-padded signals. Note that by computing the FFT of a zero-padded signal, we are effectively performing sinc interpolation in the frequency domain for the shorter sequence. The magnitudes of the spectrums for two1 and two2 are shown in figure 7.3.

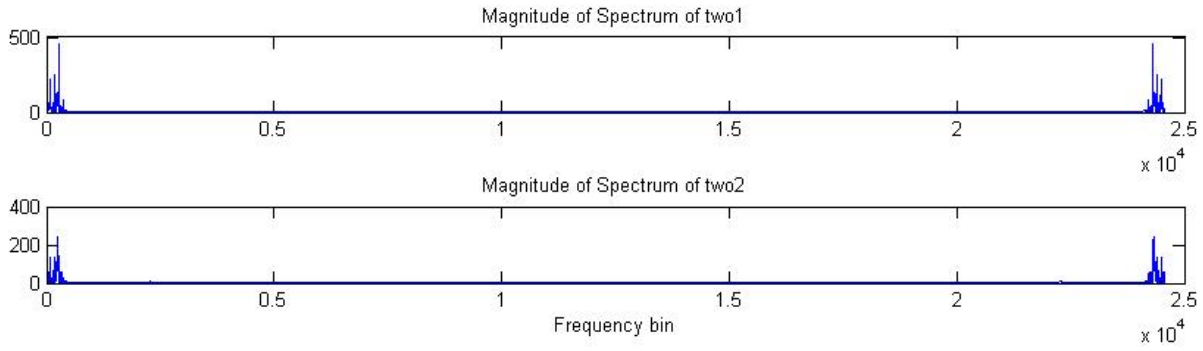

Figure 7.3: Magnitude of the spectrums of two1 and two2 signals.

In figure 7.4, we zoom into the chart for improved resolution. We also show the spectrum of Nicholas saying the word “one” for comparison (this signal will be called one1 in this document). By eye, we are able to see that the spectrums of two1 and two2 are more similar to each other than the spectrum of one1 is.

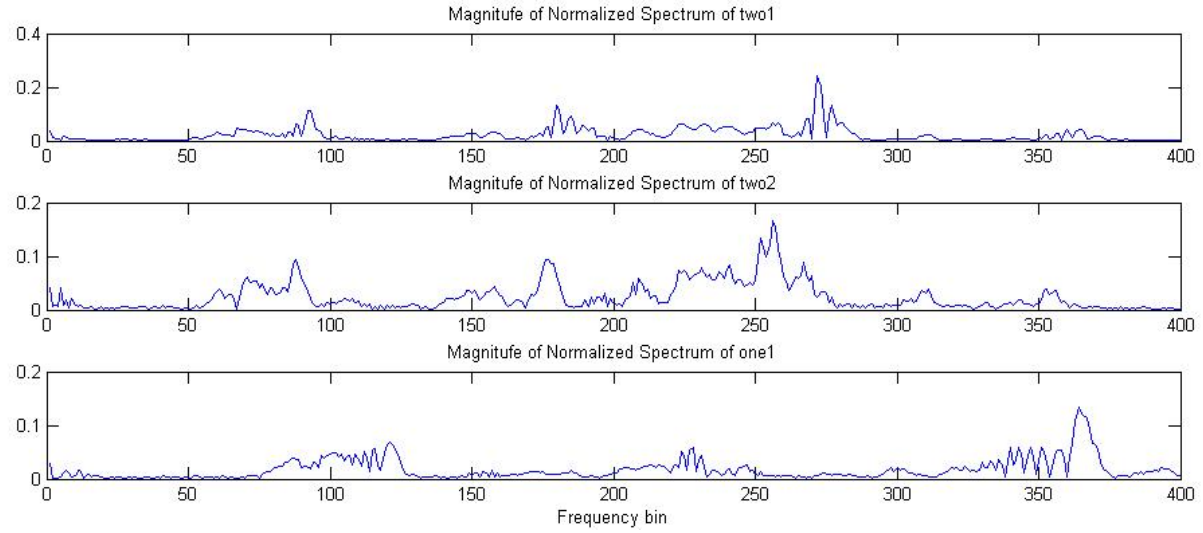

Figure 7.4: Magnitude of the low frequency portion of spectrums of signals.

After identifying the relevant spectrums, we normalize them by the amount of energy in the spectrum. That is, we convert them according to the formula shown in (6.3), this time the data in question is the signal’s

spectrum. According to Parseval's theorem, this is equivalent to performing normalization in the time domain (of the zero padded signals). Figure 7.5 shows the normalized spectrums.

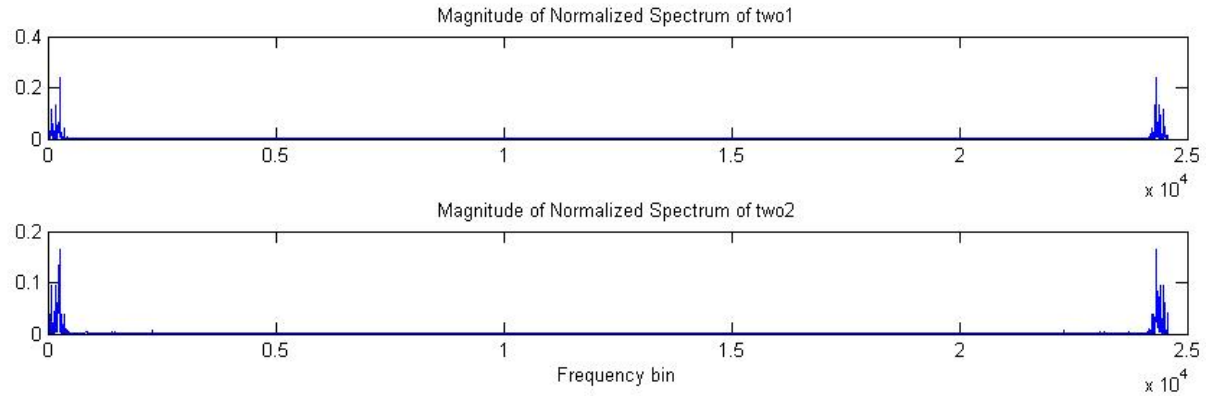

Figure 7.5: Magnitude of normalized spectrums for two1 and two2 signals.

Finally, we compute the element by element difference of the two spectrums and calculate the norm of this difference. For the "two" normalized spectrums used in this example, the norm of the difference was approximately 143%. After normalization, the energy in each individual spectrum is 1. Again, the energy in the difference signal is very high for two signals that are the "same".

We begin to understand why when we statistically analyze a set of recordings of the same phrase. Shown in figure 7.6 is a set of spectrums for recordings made by Nicholas stating the word "one". By observing these plots, we gain some intuition into what a word is. Let us call each bump in the spectrum a "pocket" of energy. We see that the word "one" has five pockets of energy. We see that different recordings have pockets located at approximately the same frequency bins, but that the shape of each pocket is different. Because of this difference, there is variability in the average of frequency bins with high average energy values. This variability is quantified through the standard deviation, shown in the last row of plot 7.6.

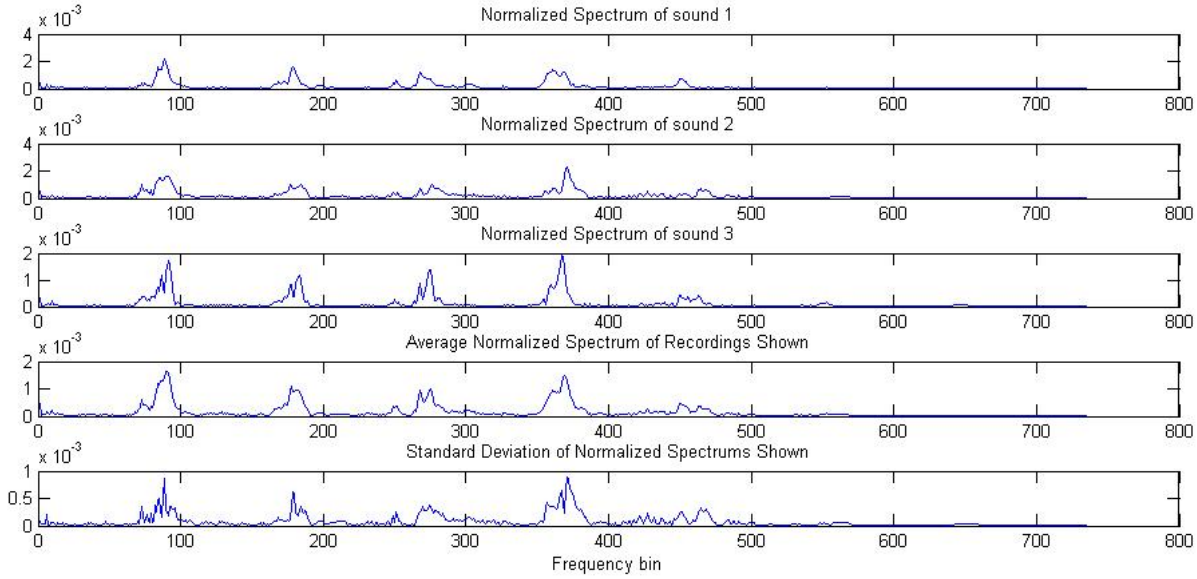

Figure 7.6: A depiction of spectrum magnitudes of several recordings of Nicholas saying the word "one", the average and standard deviation of those spectrums.

We can use the L2 norm as a measure of the difference between the spectrum of two signals. We compute the L2 norm of Nicholas stating the word one (spectrum not shown) and the average "one" spectrum – the value is approximately 54%. When making the same comparison between the average spectrum and Nicholas stating the word "two", the value becomes 75% - a percentage difference of approximately 36%. The deviation between the average "one" spectrum and Matt stating the word "one" is 62%.

We can take advantage of the knowledge of the variability of the signal in our comparison metric. To account for this variability, we can use a weighted L2 norm as our comparison metric. We define our weighted norm in (7.2).

	(7.2)

$$c = \sum_{i}^{\min(N_1,N_2)} \frac{\left(f(i)-\bar{d}(i)\right)^2}{\left(2\times10^{-4}\right)+\sigma(i)}$$

This metric reduces the importance of mean data values with high variance, and increases the penalty for data values with low variance. We include an addition to a constant in the denominator to prevent division by 0, and we set this constant equal to 2×10^{-4} since we notice that the noise in the normalized spectrum is around this level.

Using the weighted norm, we calculate a comparison metric between the average spectrum shown in figure 7.6, and a separate recording of Nicholas saying the word “one” to be 452. This seems like a high number, but it is no longer a physical quantity. We compare this value to the metric determined between the average spectrum shown in figure 7.6 and Nicholas saying the word “two”: 656. Notice that the difference in metric values is approximately 45%. The weighted norm value between the average spectrum shown in figure 7.6 and Matthew stating the word “one” is 691; a difference of approximately 53%.

Unfortunately, since the weighted norm value is not a physical quantity, we would require a large database of signals to determine the appropriate value for our threshold. In lieu of this, we will continue to use the L2 norm as our comparison metric.

VIII – Results

Table 8.1 shows the values of the L2 norms against a database of sounds generated by Nicholas. The first row represents a comparison of phrases generated by Nicholas to the database. The second row represents the same phrases generated by Andrew to the database.

	“one”	“two”	“three”	“four”	“five”	“six”	“seven”	“eight”	“nir
Nicholas	0.37	0.27	0.18	0.28	0.24	0.21	0.31	0.29	0.4(
Andrew	0.78	0.67	0.66	0.74	0.39	0.50	0.64	0.62	0.6⁴

Table 8.1: L2 Norm values for comparison to a database of Nicholas signals.

It is highly likely that the two distributions represent different quantities. The mean and standard deviation for the L2 norm values generated by Nicholas (Andrew) are 0.3 and 0.08 (0.63 and 0.11). This shows that by placing a threshold at around 0.45, we can separate the database into matches and impostors.

We can make a similar comparison to the database of sounds generated by Andrew; table 8.2 shows the values of L2 norms against this database.

	“one”	“two”	“three”	“four”	“five”	“six”	“seven”	“eight”	“nir
Andrew	0.23	0.45	0.31	0.32	0.41	0.28	0.24	0.35	0.4
Nicholas	0.76	0.63	0.58	0.51	0.48	0.41	0.51	0.62	0.5

Table 8.2: L2 Norm values for comparison to a database of Andrew signals.

Note that the populations L2 norm values from the true match (Andrew) and the impostor (Nicholas) are not as disparate as the analgous values when comparing to the database generated by Nicholas. In fact, the L2 norm for the true “nine” match is actually higher than the impostor “six” match.

This is not completely unexpected. Indeed, the same words issued by different people have great similarities – thus they are understood to have the same meaning. We can reduce the probability of false alarm (the probability that we will inappropriately mark a true match as an impostor) by combining the results from the matches of all four components of the Personal Identification Number.

Specifically, we are able to set the threshold for the L2 norm to a high value (e.g. 0.48). Then, if the comparison value exceeds this amount for any one phrase, the candidate is flagged as an impostor.

Figure 8.1 shows the Receiver Operating Characterisic (ROC) curve for the algorithm. This curve was generated by varying the L2 norm threshold used distinguish matches from non-matches. For this plot, a detection is an

accurate classification of an impostor as an impostor. A false alarm is a classification of a valid user as an impostor.

We generated this curve using databases of eight recordings per signal. Additionally, for this ROC curve, impostors were always implemented with the correct PIN numbers.

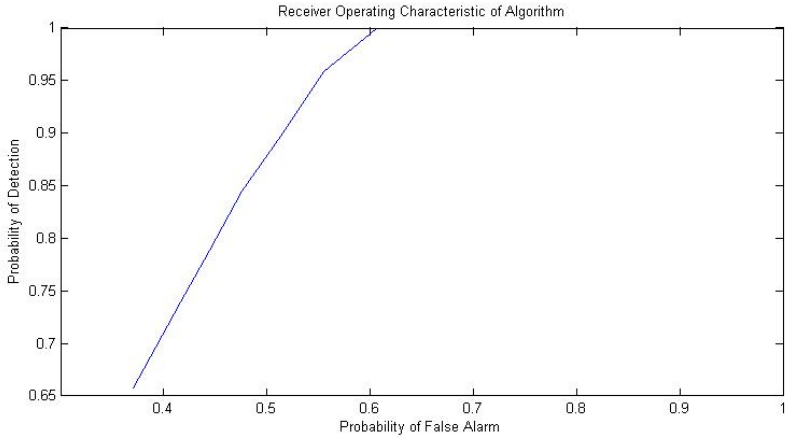

Figure 8.1: Receiver Operating Characteristic of complete algorithm

For a probability of 50% to accurately identify a valid user as a match, the probability of detecting an impostor as an impostor is approximately 88%. This corresponds to a L2 norm threshold value of 0.70. At this operating level, on average, a valid user would have to repeat his/her pin twice before being correctly identified as a match. Increasing the probability of detecting an impostor to 95% only increases the probability of false alarm to approximately 55%. This corresponds to a threshold value of 0.65.

Summary (Analysis of Speech Signal Spectrums Using the L2 Norm)
Summary

IX – Summary

We have created an algorithm that makes comparisons between a stored Personal Identification Number (PIN) and a candidate PIN. The comparison is the L2 norm of the difference of the spectrum of the signals.

Download Matlab Files (Analysis of Speech Signal Spectrums Using the L2 Norm)
Download Matlab files for this project here.

The Matlab files are a *.zip files attached to this module. To download them, click on the 'Metadata' link below.